Web Scraping and Data Analysis Mini Project

Abstract

This project demonstrates the integration of web scraping and data analysis through Python's widely-used libraries. It involves extracting data from web sources, saving the data locally as CSV, re-uploading it for cleaning, and performing exploratory data analysis. The project is implemented in **Google Colab** and focuses on ensuring a modular and reusable workflow. By separating the web scraping and data cleaning stages, this mini-project showcases a streamlined, end-to-end pipeline for gathering and analyzing real-world data in a reproducible manner.

Acronyms

- **CSV**: Comma-Separated Values
- **HTTP**: Hypertext Transfer Protocol
- **EDA**: Exploratory Data Analysis
- **HTML**: Hypertext Markup Language
- **Pandas**: Python Data Analysis Library
- **API**: Application Programming Interface

1. Project Properties

- **Project Name**: Web Scraping and Data Analysis Mini Project
- **Author :** K.KAMALESH
- **Software Used**: Google Colab, Python (Version 3.x)
- **Primary Libraries**:
    - requests (HTTP Requests)
    - beautifulsoup4 (HTML Parsing)
    - pandas (Data Manipulation)
- **Programming Language**: Python
- **File Format Used**: CSV (for storing and re-uploading scraped data)

---

2. Detailed Project Description

2.1 Overview

This mini-project is a combination of web scraping and data analysis techniques. The goal of the project is to scrape data from a target website, store it locally, re-upload it for data cleaning, and then carry out data analysis using Python's rich ecosystem of data processing libraries.

2.2 Tools and Libraries

- **Google Colab**: A cloud-based environment for Python execution, enabling the user to run the entire project seamlessly.
- **Python Libraries**:
    - requests: To send GET requests and fetch data from the web.
    - BeautifulSoup: A parsing tool for HTML content to extract meaningful data.
    - pandas: For handling tabular data (stored in CSV format) and performing data cleaning, transformation, and analysis.

2.3 Workflow

1. **Web Scraping**:
    - The process begins by sending HTTP requests using the requests library to fetch HTML content from the website.
    - BeautifulSoup is used to parse the HTML and extract the relevant data (e.g., text, tables, images, or lists).
2. **Data Storage**:
    - Once the data is scraped, it is saved in CSV format locally using the pandas library. The CSV format ensures easy re-usability of the data.
3. **Re-uploading for Data Cleaning**:
    - The locally saved CSV file is re-uploaded into the Google Colab environment for cleaning.
    - During cleaning, operations such as removing null values, dealing with duplicates, and fixing formatting issues are performed using pandas.
4. **Data Analysis**:
    - After cleaning, exploratory data analysis (EDA) is carried out. This involves filtering the data, calculating statistics, and visualizing key insights.
    - Grouping, filtering, and visualizing data patterns are key steps in this phase.

2.4 Features

- **Separation of Concerns**: The project emphasizes the separation of web scraping and data cleaning stages by saving and re-uploading the CSV. This ensures a more modular and robust approach, allowing easy reprocessing of the data even if the source website is no longer available.
- **Modularity**: Each part of the workflow (scraping, cleaning, analysis) is designed to be independently modifiable.
- **Reproducibility**: By saving the extracted data in CSV format, the project ensures that the data cleaning and analysis steps can be repeated without re-running the web scraping phase.

---

3. Project StructureDetails About the CSV File

CSV Columns and Data

- Name: The name of the company.
- CMP-Rs: Current Market Price (in Indian Rupees).
- P/E: Price-to-Earnings Ratio, which indicates the company's stock price relative to its earnings.
- MarCap-Rs.Cr.: Market Capitalization (in Crores), representing the total market value of the company.
- DivYld%: Dividend Yield, the percentage return from dividends relative to the stock price.
- NPQtr-Rs.Cr.: Net Profit of the company for the quarter (in Crores).
- QtrProfitVar-%: Percentage variation in profit from the previous quarter.
- SalesQtr-Rs.Cr.: Sales of the company in the last quarter (in Crores).
- QtrSalesVar-%: Percentage variation in sales from the previous quarter.
- ROCE-%: Return on Capital Employed, a measure of a company's profitability.
- Debt/Eq: Debt-to-Equity Ratio, indicating the company's financial leverage.
- NPPrevAnn-Rs.Cr.: Net Profit of the company for the previous annual period (in Crores).

Source Website Details

The data in this CSV was scraped from a financial market website that provides details about the stock prices, earnings, and financial performance of companies listed in India. The data includes essential metrics such as market capitalization, profit, return on capital, and sales figures, which are key indicators for financial analysis.

The web scraping process involved:

1. Sending HTTP requests to the target website to retrieve HTML content.
2. Parsing the HTML with BeautifulSoup to extract tables containing the required financial data.
3. Saving the extracted data into a CSV file for further analysis.

Data Cleaning Steps

The raw data scraped from the website often contains inconsistencies, so the following cleaning steps were performed:

1. Handling Missing Data:
    - Any rows with significant missing values were reviewed.
    - Missing numerical values were either filled using mean or median imputation or dropped if they were not critical.
2. Removing Duplicates:
    - Duplicate rows, if any, were removed to ensure the uniqueness of each company entry.
3. Correcting Data Types:

    - Ensured numerical columns such as CMP-Rs, MarCap-Rs.Cr., and DivYld% were correctly formatted as floats for accurate calculations.
    - Converted categorical values, where applicable (e.g., company names), into string types.
4. Normalizing Data:
    - Ensured consistent naming and units across all columns (e.g., ensuring all values were in Crores for financial figures).
5. Outlier Detection:
    - Reviewed extreme values, especially in financial metrics like P/E and Debt/Eq, to identify any data anomalies or inconsistencies.

Final Data Used

After cleaning, the data was well-prepared for exploratory data analysis (EDA), which included:

- Calculating the average market capitalization and comparing companies' performance across different financial metrics.
- Visualizing trends, such as the relationship between P/E ratio and ROCE to analyze profitability.

Visualizations and Their Purposes

1. CMP vs Market Capitalization

- **Plot Type**: Scatter Plot
- **Purpose**: This scatter plot visualizes the relationship between the **Current Market Price (CMP)** and **Market Capitalization** of companies. It helps analyze whether higher CMP values correspond to larger market capitalizations and identifies outliers, revealing companies with disproportionate stock prices or market values.

---

2. Current Price of Companies

- **Plot Type**: Bar Plot
- **Purpose**: This bar plot compares the **Current Market Price (CMP)** of different companies. It allows easy identification of the most and least expensive stocks, helping investors gauge market trends and company stock pricing at a glance.

---

3. Current Price vs. Price-to-Earnings (P/E) Ratio

- **Plot Type**: Scatter Plot

- **Purpose**: This scatter plot shows how the **Current Market Price (CMP)** relates to the **Price-to-Earnings (P/E) Ratio** for each company. It provides insight into the valuation of companies based on their earnings, helping to identify under- or over-valued stocks.

---

4. Sales Variation vs Profit Variation

- **Plot Type**: Bar Plot
- **Purpose**: This bar plot highlights the relationship between **Quarterly Sales Variation** and **Quarterly Profit Variation**. It helps assess whether changes in sales directly affect profit, offering insights into company performance trends over recent quarters.

---

4. Acronyms and Terminology

- **CSV**: A format for storing tabular data, where values are separated by commas.
- **HTTP**: Protocol used for sending requests and receiving data over the internet.
- **EDA**: A process in data analysis that involves summarizing the main characteristics of the data.
- **HTML**: The standard language for creating web pages, from which data is extracted.
- **Pandas**: A Python library that provides data structures and data analysis tools.
- **API**: A set of rules that allows one application to interact with another.

---

5. Conclusion

This mini-project showcases a comprehensive approach to data extraction and analysis by integrating web scraping and data cleaning in a structured, reproducible workflow. By using Python's powerful libraries and Google Colab's interactive environment, the project demonstrates the benefits of modularity, reproducibility, and clarity in handling real-world data.

4. Project Resources

- **Source Code**:Github link
- **Page Source :** Screener